CS F364: Design & Analysis of Algorithms
Mid-Semester Examination

BITS Pilani, Hyderabad Campus

June 20, 2026

Time: 90 minutes — Max Marks: 90

Instructions: Answer all six questions. Each question carries marks as indicated. Show all steps for full credit. Unsupported answers receive no credit. Closed book. No electronic devices.

1. (15 points) **Asymptotic Notation**

   (a) (5 points) Arrange the following functions in **increasing order** of growth rate (slowest-growing first):

   $$f_1(n) = n^2, \quad f_2(n) = n!, \quad f_3(n) = n, \quad f_4(n) = \sqrt{n}, \quad f_5(n) = 2^n, \quad f_6(n) = n \log n$$

   (b) (5 points) For each function below, select the **tightest** Big-Oh bound $O(f(n))$ from the list:

   $$\{O(1), O(\log n), O(n), O(n \log n), O(n^{1.5}), O(n^2), O(2^n), O(n^{105})\}$$

   i. $500n + 5n^{1.5} + 50n \log n$
   ii. $n^2 \log n + n(\log n)^2$
   iii. $3 \log_8 n + \log_2 \log_2 \log_2 n$
   iv. $2^n + n^{105} + 0.5n^{1.25}$

   (c) (5 points) Prove the following statement **from the definition** of Big-Oh:

   $$2^{n+5} = O(2^n)$$

2. (20 points) **Recurrences — Master Theorem**

   Prove the Master Theorem using the recursion tree method by the following steps:

   (a) (5 points) State the three cases of the Master Theorem for recurrences of the form $T(n) = aT(n/b) + f(n)$, where $a \geq 1$ and $b > 1$. For each case, specify the relationship between $f(n)$ and $n^{\log_b a}$ and give the resulting asymptotic bound for $T(n)$.

   (b) (5 points) Analyze the recursion tree for $T(n) = aT(n/b) + f(n)$. Determine: (i) the number of subproblems at level k (root = level 0), (ii) the size of each subproblem at level k, and (iii) the total cost contributed by level k.

   (c) (5 points) Determine: (i) the number of levels in the tree, (ii) the number of leaf nodes, and (iii) the total cost contributed by the leaf level.

(d) (5 points) Express the total cost as a summation over all levels (internal + leaf). Assuming the case where $f(n) = \Theta(n^{\log_b a})$, evaluate the sum and derive the asymptotic bound for $T(n)$.

3. (10 points) **Boat Rescue**
You are a rescue coordinator on a sinking island. There are n people, each with weight w_i. You have an unlimited number of identical boats, each with a maximum weight capacity of W. Each boat can carry **at most 2 people**. Your goal is to rescue everyone using the **minimum number of boats**.

   (a) (5 points) Describe an algorithm that solves this problem. Write it in 3–4 clear steps or pseudocode.

   (b) (5 points) Apply your algorithm to the following instance. Show each step.

$$n = 5, \quad \text{weights: } [30, 50, 60, 70, 80], \quad W = 100$$

   How many boats does your algorithm use?

4. (15 points) **Majority Element**
Given an array $A[1..n]$ of integers, an element x is called a **majority element** if it appears **more than** $n/2$ times. Not every array has a majority element.

   (a) (8 points) Design an algorithm that returns a majority element if one exists, and returns `None` otherwise. Describe your algorithm in 2–3 clear sentences. Then trace your algorithm on the array $A = [2, 3, 3, 2, 3, 3, 3]$, showing the recursive calls and the result at each level.

   (b) (7 points) Write the recurrence relation for your algorithm's worst-case running time. [3 marks] Solve it using the Master Theorem. [4 marks]

5. (15 points) **Longest Common Subsequence**
Given two strings $X[1..m]$ and $Y[1..n]$, the Longest Common Subsequence (LCS) problem asks for the length and structure of the maximum length subsequence common to both.

   (a) (6 points) **Methodology:** Define the subproblems $c[i, j]$ and provide the complete recurrence relation, including base cases. State the time and space complexity of your algorithm.

   (b) (5 points) **Table Construction:** For strings X = ABCD and Y = AXCD, complete the DP table below:

$c[i,j]$	$j=0$ ($\emptyset$)	$j=1$ (**A**)	$j=2$ (**X**)	$j=3$ (**C**)	$j=4$ (**D**)
	$\emptyset$	**A**	**X**	**C**	**D**
$i=0$ ($\emptyset$)	0	0	0	0	0
$i=1$ (**A**)	0				
$i=2$ (**B**)	0				
$i=3$ (**C**)	0				
$i=4$ (**D**)	0				

   (c) (4 points) **Traceback:** Perform a traceback on your filled DP table to identify the LCS. Clearly show your path, state the final LCS string, and report the length of the result.

6. (15 points) **Data Compression - Huffman Code**
The text below is from a well-known pop culture reference:

`maytheforcebewithyou`

(a) (3 points) Count the frequency of each distinct letter in the text. How many total characters does the string contain and what are their frequencies? Answer your results in the below format:

Letter														
Freq														

(b) (7 points) Construct an optimal prefix-free binary code for these characters using Huffman code. Show the initial sorted frequencies, each merge operation (which two nodes were merged and the resulting parent's frequency), and the final tree structure.

(c) (5 points) Assign binary codes (left branch = 0, right branch = 1) to each letter and list them in a table. [2 marks] Encode the original string `"maytheforcebewithyou"` using your code and state the total number of bits. [3 marks]